

Système d'exploitation II

TP 1 : Commandes Linux et Création des processus

Le présent TP a pour objectif la mise à niveau en commandes linux et présenter les concepts de base pour la création et la manipulation des processus. Ainsi l'élève ingénieur sera mené à implémenter des processus via les appelles systèmes basiques avec le langage C.

Partie I : Manipulation des commandes Linux (Rappels) :

Dans cette partie nous proposons la manipulation des commandes les plus utilisées pour la gestion de fichier et le parcours des répertoires sous linux.

Question 1 : Utilisation du manuel des commandes.

Question 2 : Accéder à l'éditeur de commande et placer vous à la racine.

a) Afficher l'arborescence.
b) Citer le contenu de chaque dossier sous la racine « / ».

Question 3 : Création des répertoires et des fichiers

a) Accéder le répertoire home et créer le dossier TP1.
b) Sous le ce dossier créer deux dossier dont les noms sont « CODE » et « EXE ».
c) Afficher le contenu du dossier TP1. Mettez le résultat de la commande dans un fichier intitulé « fich1 ». sous le répertoire TP1.
d) Afficher le contenu du répertoire home avec les informations de chaque fichier. Mettez le résultat « fich1 » sans écraser le contenu.
e) Afficher le contenu de « fich1 » via la commande « cat ». Comment éditer un fichier ?

Question 4 : Gestions des permissions

a) Accéder le dossier « CODE ». Créer un fichier « prog.c ».
b) Visualiser les permissions attribuées au fichier « prog.c ».
c) Expliquer les champs de la commande « chmod ». Exemple **chmod 777**.
d) Rendre les permissions du fichier « prog.c » ouvert à tous les utilisateurs.

Partie II : Compilation des programme C sous linux :

a) Editer le fichier « prog.c ». Ecrire un programme qui permet de calculer l'addition de deux entiers entrés en clavier.
b) Compiler ? Lister les fichiers sous le dossier « CODE ». Qu'est ce que vous remarquer ?
c) Exécuter le programme.
d) Créer un deuxième programme « prog2.c », qui permet de convertir le nombre de seconde entré en clavier en minute.
e) Compiler prog2.c, avant de lister les fichiers sous « CODE ». Qu'est ce que vous remarquez ?
f) Pour résoudre ce problème compiler prog2.c en renomment l'exécutable par prog2.

Partie III : Création des processus :

Dans cette partie nous proposons d'implémenter un petit programme qui permet la création d'un processus avant d'afficher ces différentes informations. Noter que la commande « **ps** » permet d'afficher la table des processus sous Linux.

a) Qu'elle est la primitive qui permet la création d'un processus. qu'est ce qu'elle prend en sortie ? qu'elles sont les valeurs de retour.
b) saisir le programme suivant dans le fichier prog2.c

```
#include <sys/times.h>
        main(){
        printf (« je suis le processus père\n ») ;
        printf (« création du processus fils\n ») ;
        int p = fork() ;
                        if(p==0)
                        { printf (“je suis le processus père\n”) ; }
                        else
                        {   if (p<0)
                              { printf (« erreur de création \n») ;}
                            else
                              { printf (“je suis le processus père\n”) ; }
                        }
            }
```

c) Identifier la partie de l'exécution de chaque processus.
d) Afficher la table des processus. mettez le contenu dans le fichier « ex.txt »
e) Exécuter et compiler le programme prog2.c
f) Afficher la table des processus. Mettre le contenu dans fichier « ex1.txt ». afficher la différence entre les deux fichiers. Qu'est ce que vous remarquez ?

Annexe : Tableau des commandes

Commande	Description
cd	Accéder à un dossier
ls	Afficher le contenu d'un répertoire
man	Afficher l'aide pour une commande ou l'ensemble des commandes
touch	Création d'un fichier
mkdir	Création d'un répertoire
gcc	Compilation d'un programme C
cat	Visualiser un fichier
diff	afficher les différences entre 2 fichiers.
Ps -ef	Afficher la table des processus
chmod	Changer les permissions